P.D. 00/05/97 p. 583-595 = (13)

10/756165 U.S. PTO 19270 011204

A Novel Block Truncation Coding of Color Images Using a Quaternion-Moment-Preserving Principle

Soo-Chang Pei, *Senior Member, IEEE*, and Ching-Min Cheng, *Member, IEEE*

***Abstract*— Block truncation coding (BTC) is an efficient tool for image compression. To compress color-pixel blocks, a novel color BTC algorithm, called quaternion-moment block truncation coding (QMBTC), is presented in this paper. Analytical formulas for QMBTC, whose computation time is on the order of pixel block size, are derived by using quaternion arithmetic and the moment-preserving principle. The proposed color BTC algorithm can adaptively truncate a pixel block into one or two output classes according to the distribution of color values inside the blocks. The experimental results show that the compression ratio is increased as compared with existing color BTC algorithms, and the picture quality of reconstructed images is satisfactory. In addition, a post-BTC data compression scheme is proposed to further compress the subimage constructed by reproduction colors of truncated pixel blocks. Using a lookup table to display decoded data, this postprocessing scheme can output images acceptable to human eyes.**

***Index Terms*—Block truncation coding, color image compression, quaternion-moment-preserving.**

I. Introduction

BLOCK truncation coding (BTC) was first proposed by Delp and Mitchell [1], [2] to compress monochrome images. Unlike other image compression methods such as transform coding and vector quantization [3], BTC requires less computation effort and has good capability of combating channel errors. Lema and Mitchell also proposed a monochrome BTC algorithm [4] by preserving the mean-absolute moment or minimizing the mean-absolute error in each pixel block. The outputs of these BTC quantizers for each pixel block include two numbers which are the associated centroids of two partitioned classes and a bit map which specifies the states of each output pixel. The bit map (1 bit/pixel) occupies more than 61% of the output code. Although some efforts have been made to compress the bit map [5], [6], the subjective results are not quite satisfactory.

The BTC method described above can be extended to color images by applying the BTC technique to each of the color planes [4]. Based on empirical results [4], good quality images at 6 bits/pixel have been reported. Since BTC is a two-level quantizer that adapts to local properties of the image, the three resulting bit maps will be quite similar or almost identical. This motivates the use of one bit map to quantize all three of the color planes. Thus, only one out of three bit maps needs to be stored. Several single bit-map color BTC algorithms [7]–[9] have been proposed. The color cell compression (CCC) algorithm, which is presented by Campbell *et al.* [7], uses the sample mean value of the luminance values of the pixel block to truncate the color pixels inside the block into two classes. Then the centroids of each class are chosen as the reproduction colors of the block. Wu and Coll's algorithm [8] truncates the average of the R, G, B planes for the corresponding pixels by choosing the sample mean value as the threshold. Recently, Kurita and Otsu proposed a color BTC algorithm [9], which is based on truncating the principal component of R, G, and B values inside the pixel block. They reported that the mean vector is preserved, and the covariance matrix of the decoded block is the same as the between class covariance matrix. When no post-BTC data compression scheme is employed, these techniques [7]–[9] can obtain the resultant bit rate of 4 bits/pixel for the 4×4 block size.

In contrast to the above-mentioned color BTC algorithms, which process on only one transformed component of input color data, we propose in this paper a novel color BTC algorithm, called quaternion-moment block truncation coding (QMBTC). QMBTC generalizes conventional monochrome BTC [1] to color BTC by expressing the input color space as a quaternion-valued space. Through the definition of quaternion moments of input color data, we extend the moment-preserving principle introduced by Delp and Mitchell [1] from one-dimensional (1-D) monochrome data to three-dimensional (3-D) color data. An analytic solution for QMBTC is also obtained by the usage of quaternion arithmetic. One feature of QMBTC is that it can determine the number of output classes according to the distribution of color values inside the pixel block. A pixel block with similar color values produces only one output class, and the associated bit map can be replaced by a 1 bit reference indicating that one color clustering happened. Thus, QMBTC can achieve a better compression ratio than existing color BTC algorithms. As well, with no severe performance degradation, a post-BTC data compression method is introduced to further compress color images.

This paper is organized as follows. Some preliminaries of quaternion arithmetic and the definition of quaternion moments are first introduced in Section II. Section III presents the QMBTC and its performance evaluation. In Section IV, the application of the proposed single bit map color BTC for image

Paper approved by H. Watanabe, the Editor for Video/Image Communications of the IEEE Communications Society. Manuscript received November 27, 1995; revised April 10, 1996. This work was supported by the National Science Council, R.O.C. under Contract NSC 85-2213-E-002-025. This paper was presented in part at IEEE ISCAS'96, Atlanta, GA, May 1996.

S.-C. Pei is with the Department of Electrical Engineering, National Taiwan University, Taiwan, R.O.C. (e-mail:pei@cc.ee.ntu.edu.tw).

C.-M. Cheng is with the Telecommunication Laboratory, Ministry of Communications, Taiwan, R.O.C. (e-mail:cmc@ms.tl.gov.tw).

Publisher Item Identifier S 0090-6778/97/03718-5.

compression is addressed, including the implementation of a post-BTC data compression method. In Section V, we compare the performance of existing color BTC algorithms with the proposed algorithm. Finally, some conclusions are made in Section VI.

II. Quaternion Moments

Human perception of gray level images is quite satisfactory, but color images do seem to be perceptually richer. And, different from gray level, color data representations are usually ternary. For example, color image data from a frame grabber are usually in the RGB (red, green, blue) primary color space. There are other colorimetric representations which allow the entire visual gamut being described. Some of the examples are: CIE1930-XYZ, CIE1976-$L^*a^*b^*$, HSV, and CCIR601-YCrCb color coordinate spaces [3]. To effectively compute multidimensional color data, an efficient expression of color data is necessary. Machuca and Phillips [10] have proposed vector fields as the theoretical model of color data. Through the usage of differential geometry and vector analysis, they obtained the algorithm for color edge detection.

In contrast to using vector fields [10], we select an algebraic approach in this paper to generalize the moment-preserving principle of monochrome BTC [1] to multidimensional color image. Although real and complex number systems are used to provide arithmetic operations of one-dimensional and two-dimensional data, there are no usual laws of arithmetic to handle algebraic operations of ternary numbers. We choose the quaternion number, which was discovered by Hamilton [11], as the expression of color data. In this expression, one dimension of the quaternion number is redundant and set to zero. The algebra of the quaternion number, quaternion arithmetic, is the generalization of complex numbers. Using quaternion arithmetic, analytical formulas for monochrome BTC [1] can still be maintained for a color image. Routine BTC [1] is a special case of the proposed moment-preserving BTC in the quaternion space. In this section, we will address some preliminaries of quaternion arithmetic and the definition of quaternion moments used by the following sections.

Considering a 4-D real-valued data set H $= \{(q_0(n), q_1(n), q_2(n), q_3(n))\}_{n=1}^{N}$, a quadruple data point $(q_0(n), q_1(n), q_2(n), q_3(n))$ can be expressed as a quaternion number $\hat{q}(n)$

$$\hat{q}(n) = q_0(n) + q_1(n)\cdot i + q_2(n)\cdot j + q_3(n)\cdot k \tag{1}$$

where i, j, and k denote the operation units of quaternion number. Any vector $\boldsymbol{v} \in R^3$ can be expressed as a quaternion with q_0 set to be zero. For example, a color value (R, G, B) can be shown as a quaternion with $q_1 = R, q_2 = G, q_3 = B, q_0 = 0$. Any vector $\boldsymbol{v} \in R^2$ can be expressed like a complex number. A quaternion can also be denoted as $\hat{q}(n) = \langle \boldsymbol{a}, b\rangle$ where $\boldsymbol{a} = (q_1(n), q_2(n), q_3(n))$ and $b = q_0(n)$.

Operations on quaternion number have the following properties.

1) The addition and subtraction rules of quaternions are the same as for complex numbers.

2) Using the cross product of vector space $(\times)$, one can define multiplication of two quaternions, $\hat{q}$ and $\hat{q}'$ as

$$\begin{aligned}\hat{q}\cdot\hat{q}' &= \langle \boldsymbol{a}, b\rangle \cdot \langle \boldsymbol{a}', b'\rangle \\ &= \langle \boldsymbol{a}\times\boldsymbol{a}' + b\cdot\boldsymbol{a}' + b'\cdot\boldsymbol{a}, b\cdot b' - \boldsymbol{a}\cdot\boldsymbol{a}'\rangle.\end{aligned} \tag{2}$$

3) The conjugate $\hat{q}^*$ of $\hat{q}$ is defined as

$$\hat{q}^* = -\langle \boldsymbol{a}, b\rangle = q_0 - (q_1\cdot i + q_2\cdot j + q_3\cdot k) \tag{3}$$

and the norm of the quaternion is denoted as $\|\hat{q}\|^2 = \hat{q}\cdot\hat{q}^*$.

4) The reciprocal of $\hat{q}$ is

$$(\hat{q})^{-1} = \frac{\hat{q}^*}{\|\hat{q}\|^2}. \tag{4}$$

With the help of the $(\hat{q})^{-1}$, the division of quaternions is denoted as

$$\frac{\hat{q}'}{\hat{q}} = \hat{q}'\cdot(\hat{q})^{-1}. \tag{5}$$

Based on the above definition of the quaternion, we will designate the first-, second-, and third-order quaternion moments as follows in order to explicitly express the statistical parameters of a 4-D data point:

$$\begin{aligned}\hat{m}_1 &= E[\hat{q}] \\ \hat{m}_2 &= E[\hat{q}\cdot\hat{q}^*] \\ \hat{m}_3 &= E[\hat{q}\cdot\hat{q}^*\cdot\hat{q}]\end{aligned} \tag{6}$$

where $E[\cdot]$ represents the expectation. The definitions of $\hat{m}_1$ and $\hat{m}_2$ are the extension of complex moments. The definition of the third-order quaternion moment $\hat{m}_3$ is adopted from high-order statistics [12]. Equation(6) can be further expressed as

$$\begin{aligned}\hat{m}_1 &= E[q_0] + E[q_1]\cdot i + E[q_2]\cdot j + E[q_3]\cdot k \\ \hat{m}_2 &= E[q_0^2 + q_1^2 + q_2^2 + q_3^2] \\ \hat{m}_3 &= E[q_0^3 + q_1^2 q_0 + q_2^2 q_0 + q_3^2 q_0] \\ &\quad + E[q_1^3 + q_0^2 q_1 + q_2^2 q_1 + q_3^2 q_1]\cdot i \\ &\quad + E[q_2^3 + q_0^2 q_2 + q_1^2 q_2 + q_3^2 q_2]\cdot j \\ &\quad + E[q_3^3 + q_0^2 q_3 + q_1^2 q_3 + q_2^2 q_3]\cdot k.\end{aligned} \tag{7}$$

To calculate the quaternion moments of the data set H, the expectation $E[\cdot]$ of (7) is replaced by the sample mean value. As we observe, moment $\hat{m}_1$ represents the average magnitude of the input data (q_0, q_1, q_2, q_3). The real-valued $\hat{m}_2$ expresses the expected value of the power of (q_0, q_1, q_2, q_3). The third moment $\hat{m}_3$ consists of the sum of joint third-order moments among q_0, q_1, q_2, and q_3 which reflect the measure of skewness of the input data. Although statistical distribution of a color image can be described by knowing each order of moment, it is known that the higher the order of moment is, the less important this moment is. Thus, preserving moments up to third order would not severely lose statistical distribution of the image. To derive analytical formulas for QMBTC in the next section, we will use the moment-preserving principle on $\hat{m}_1, \hat{m}_2$, and $\hat{m}_3$ in order to maintain statistical characteristics of input image. As empirical results will show, the output picture quality is satisfactory for human eyes.

III. Quaternion-Moment Block Truncation Coding

The problem of applying QMBTC in a pixel block H is to sel ct a hyperplane as a threshold such that, if all below-threshold pixels in H are replaced by the quaternion value $\hat{z}_0$ and those above-threshold pixels in H are set to be the quaternion value $\hat{z}_1$, then the first three quaternion moments of image H are preserved in the resultant two-level pixel block G.

Let p_0 and p_1 denote the probabilities of the below-threshold and the above-threshold pixels in H after the QMBTC, respectively, and

$$p_0 + p_1 = 1. \tag{8}$$

Then the first three quaternion moments of the two-level pixel block G are given by

$$\begin{aligned}\hat{m}'_1 &= \sum_{l=0}^{1} p_l \cdot \hat{z}_l \\ \hat{m}'_2 &= \sum_{l=0}^{1} p_l \cdot \hat{z}_l \cdot \hat{z}_l^* \\ \hat{m}'_3 &= \sum_{l=0}^{1} p_l \cdot \hat{z}_l \cdot \hat{z}_l^* \cdot \hat{z}_l.\end{aligned} \tag{9}$$

If we let the first three quaternion moments of the two-level pixel block G be equal to those of H, $\hat{m}_v = \hat{m}'_v$, $v = 1, 2, 3$, we get the following moment-preserving equations:

$$\begin{aligned}p_0 \cdot \hat{z}_0 + p_1 \cdot \hat{z}_1 &= \hat{m}_1 \\ p_0 \cdot \hat{z}_0 \cdot \hat{z}_0^* + p_1 \cdot \hat{z}_1 \cdot \hat{z}_1^* &= \hat{m}_2 \\ p_0 \cdot \hat{z}_0 \cdot \hat{z}_0^* \cdot \hat{z}_0 + p_1 \cdot \hat{z}_1 \cdot \hat{z}_1^* \cdot \hat{z}_1 &= \hat{m}_3.\end{aligned} \tag{10}$$

Thus, using the quaternion moments, the Delp and Mitchell moment-preserving principle [1] can still be maintained for color image data, and the solution will be a special case of the proposed QMBTC.

The moment-preserving equations (10) can be solved indirectly by adopting the following polynomial of quaternion $\hat{z}$.

$$\begin{aligned}C(\hat{z}) &= \hat{z} \cdot \hat{z}^* + \hat{z}^* \cdot \hat{c}_1 + \hat{c}_0 \\ &= (\hat{z} - \hat{z}_0)(\hat{z} - \hat{z}_1)\end{aligned} \tag{11}$$

where $\hat{z}_l$ for $l = 0, 1$, the quantized levels of QMBTC, are the roots of $C(\hat{z})$, i.e., $C(\hat{z}_l) = 0$. If we multiply $C(\hat{z}_l)$ with (8) and the equation defining $\hat{m}'_1$, respectively, we get

$$\begin{cases} p_0 \cdot C(\hat{z}_0) + p_1 \cdot C(\hat{z}_1) = 0 \\ p_0 \cdot \hat{z}_0 \cdot C(\hat{z}_0) + p_1 \cdot \hat{z}_1 \cdot C(\hat{z}_1) = 0. \end{cases} \tag{12}$$

Then, using (10), $\hat{c}_0$ and $\hat{c}_1$ of (11) can be obtained as

$$\begin{aligned}\hat{c}_1 &= \frac{(\hat{m}_3 - \hat{m}_1 \cdot \hat{m}_2)}{(\hat{m}_1 \cdot \hat{m}_1^* - \hat{m}_2)} \\ \hat{c}_0 &= -(\hat{m}_1^* \cdot \hat{c}_1 + \hat{m}_2).\end{aligned} \tag{13}$$

With the help of $\hat{c}_0$ and $\hat{c}_1$, the roots of $C(\hat{z}_l) = 0$ can be solved.

Considering the case of thresholding color pixels with q_0 set to zero, the unknown $\hat{z}_l$ can be denoted as

$$\hat{z}_l = z_{l1} \cdot i + z_{l2} \cdot j + z_{l3} \cdot k. \tag{14}$$

If we express $\hat{c}_1$ and $\hat{c}_0$ of (13) as

$$\begin{aligned}\hat{c}_1 &= c_{11} \cdot i + c_{12} \cdot j + c_{13} \cdot k \\ \hat{c}_0 &= c_{00} + c_{01} \cdot i + c_{02} \cdot j + c_{03} \cdot k\end{aligned} \tag{15}$$

we have the following results after putting (14) and (15) into (11) and letting (11) equal zero.

$$\begin{aligned}z_{l1} &= \frac{c_{12} \cdot z_{l3} + c_{01}}{c_{13}} \\ z_{l2} &= \frac{c_{11} \cdot z_{l3} - c_{02}}{c_{13}} \\ a \cdot z_{l3}^2 &+ b \cdot z_{l3} + d = 0\end{aligned} \tag{16}$$

where

$$\begin{aligned}a &= c_{11}^2 + c_{12}^2 + c_{13}^2 \\ b &= 2 \cdot c_{01} \cdot c_{12} - 2 \cdot c_{02} \cdot c_{11} + c_{11}^2 \cdot c_{13} \\ &\quad + c_{12}^2 \cdot c_{13} + c_{13}^3 \\ d &= c_{01} \cdot c_{12} \cdot c_{13} - c_{02} \cdot c_{11} \cdot c_{13} \\ &\quad + c_{01}^2 + c_{02}^2 + c_{00} \cdot c_{13}^2.\end{aligned}$$

From (16), the two solutions z_{03} and z_{13} can be obtained directly from the quadratic formula

$$\begin{aligned}z_{03} &= \frac{-1}{2 \cdot a} \cdot (b - \sqrt{b^2 - 4 \cdot a \cdot d}) \\ z_{13} &= \frac{-1}{2 \cdot a} \cdot (b + \sqrt{b^2 - 4 \cdot a \cdot d}).\end{aligned} \tag{17}$$

The corresponding z_{l1} and z_{l2} can thus be acquired from (16). After $\hat{z}_0$ and $\hat{z}_1$ are obtained, the hyperplane l', which is perpendicular to the line segment $\overline{\hat{z}_0\hat{z}_1}$ such that the quaternion space is split into two halves, can be defined as the decision boundary to separate the input data set into two classes. The half space containing $\hat{z}_0$ has p_0 portion of pixels, and the other half space containing $\hat{z}_1$ has p_1 portion of pixels. However, if we want to implement the QMBTC in a much quicker way, l' can be chosen alternatively as a hyperplane perpendicular to and bisecting the line segment $\overline{\hat{z}_0\hat{z}_1}$. If the color value of each pixel is denoted as (I_1, I_2, I_3), the linear equation, which formulates the decision boundary l', is thus described as

$$I_1 = s - t_1 \cdot I_2 - t_2 \cdot I_3 \tag{18}$$

with

$$\begin{aligned}s &= \frac{(z_{01}^2 + z_{02}^2 + z_{03}^2 - z_{11}^2 - z_{12}^2 - z_{13}^2)}{2 \cdot (z_{01} - z_{11})} \\ t_1 &= \frac{(z_{02} - z_{12})}{(z_{01} - z_{11})} \\ t_2 &= \frac{(z_{03} - z_{13})}{(z_{01} - z_{11})}.\end{aligned}$$

This segmentation of two classes is equivalent to the nearest neighbor clustering of two centers $\hat{z}_0$ and $\hat{z}_1$. The classes thus obtained can be used to produce a single bit map for the pixel block. Besides, the computation complexity of the QMBTC is mainly dominated by the calculation of quaternion moments $\hat{m}_v$, for $v = 1, 2, 3$, whose computation time is of order N, the size of pixel block.

A. The Algorithm

In summary, the proposed color BTC algorithm can be described as follows.

1) Divide the input color image into small nonoverlapping blocks, 4×4 size in this paper.

2) Express the color value of each pixel (I_1, I_2, I_3) by the quaternion number

$$\hat{q} = q_0 + q_1 \cdot i + q_2 \cdot j + q_3 \cdot k$$

with $q_1 = I_1, q_2 = I_2, q_3 = I_3; q_0 = 0; i, j$, and k denote the operation units of the quaternion number.

3) Obtain two quantized levels $\hat{z}_0$ and $\hat{z}_1$ for each pixel block by solving the quaternion-moment-preserving equations (10).

4) Choose the hyperplane l' defined by (18) as the decision boundary to segment the pixel block into two halves.

5) Construct a class-indicating bit map such that each pixel location is coded as a "one" or a "zero," depending on whether or not the pixel resides in the half containing $\hat{z}_1$.

Instead of using $\hat{z}_0$ and $\hat{z}_1$, we select in this paper the centroid of each output class as reproduction colors of the truncated block in order to reduce the minimum mean-square error. To illustrate coding a pixel block, we select a 4×4 pixel block from an image and arrange the color values $(I_1, I_2, I_3) = (R, G, B)$ in the block as a quaternion-valued matrix $\boldsymbol{X}$. In $\boldsymbol{X}$, each element (q_0, q_1, q_2, q_3) is set to $(0, R, G, B)$ of the corresponding pixel location, as shown on the next page.

$$\hat{z}_1 = (0, 229.6, 135.5, 109.2)$$
$$\hat{z}_0 = (0, 206.1, 87.6, 9.4)$$

and the decision boundary l' is $I_1 = 697 - 2.03 \cdot I_2 - 4.24 \cdot I_3$. By deciding whether or not the value of $I_1 - 697 + 2.03 \cdot I_2 + 4.24 \cdot I_3$ is greater than zero for each pixel, we have the corresponding bit map expressed as

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

The root mean square error (RMSE) of QMBTC between the original and quantized block can be calculated to have the value 7.71. This example shows that only one output class is generated. When employing a k-means classifier [13] to truncate the same pixel block, we arbitrarily choose two pixel values as the initial values. After three iterations, we obtain two output classes which can be illustrated by the bit map

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

If each pixel inside the corresponding class is represented by its centroid, the RMSE of the k-means classifier has value 6.03. Another example $\boldsymbol{X}'$, shown on the next page, will demonstrate that QMBTC produces two output classes:

so

$$\hat{z}_1 = (0., 146.0, 85.2, 102.2)$$
$$\hat{z}_0 = (0., 14.1, 13.3, 29.3)$$

and the decision boundary l' is $I_1 = 143.2 - 0.545 \cdot I_2 - 0.552 \cdot I_3$ with the corresponding bit map expressed as

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

The RMSE of QMBTC has value 14.02. Using the k-means classifier [13] in this example, we also arbitrarily choose two pixel values as the initial values. After five iterations, we obtain the same two output classes as those of QMBTC.

From these examples, we observe that if color values of the block are similar, the QMBTC will return one output class. Otherwise, if the color values of the block are not very close, two color clusterings will be obtained. Concerning the k-means classifier, it always produces two output classes and the complexity of the k-means classifier is greater than that of QMBTC. For instance, the k-means classifier should worry about the number of iterations needed and the problem of being convergent or not. For the data size and the number of iterations being equal to N and I, respectively, the computation load of the k-means classifier is of order $I \cdot N$. In addition, unless the initial guess is well chosen, the results of the k-means classifier are easily trapped inside the local minimum, which degrades the performance of the k-means classifier.

If the k-means classifier wants to adaptively truncate a pixel block into one or two output classes, one intuitive measure is the RMSE between the original and quantized block. As we see, RMSE's of the k-means classifier for the above two examples are not different significantly. Moreover, considering a one-class block, whose pixel values are near, and a two-class block, whose pixel values are equal for each class, we can expect that the RMSE's of the k-means classifier for these blocks are very close. Thus, it is difficult to have a threshold to distinguish between the one-class and two-class block by using this measure.

B. Performance Analysis

In this subsection, we analyze the performance of the QMBTC operation for the case when the input color data set is composed of two equiprobable 3-D normal distributions. For each distribution, the components of data have equal variances. The covariance matrix $\boldsymbol{C}$ of these distributions is given by

$$\boldsymbol{C} = \begin{pmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{pmatrix}$$

where σ represents variance. The probability density function of these normal distributions are then denoted as

$$f_l(\boldsymbol{I}) = (2\pi^{3/2}|\boldsymbol{C}|^{1/2})^{-1} \exp(-\tfrac{1}{2}(\boldsymbol{I} - \boldsymbol{m}_l)^t \boldsymbol{C}^{-1}(\boldsymbol{I} - \boldsymbol{m}_l)) \quad \text{for } l = 0, 1$$

where $\boldsymbol{I} = (I_1, I_2, I_3)$ is the vector representing the color value of one data point. $\boldsymbol{m}_l = (\mu_l, \mu_l, \mu_l)$, for $l = 0, 1$, are the mean vectors for each distribution, and $|C|$ is the determinant of covariance matrix C. According to the moment theorem for normal distribution [14], quaternion moments $\tilde{m}_1, \tilde{m}_2$, and $\tilde{m}_3$ can be calculated as

$$\begin{aligned}\tilde{m}_1 &= \tfrac{1}{2}\cdot(\mu_0+\mu_1)\cdot i + \tfrac{1}{2}\cdot(\mu_0+\mu_1)\cdot j\\ &\quad + \tfrac{1}{2}\cdot(\mu_0+\mu_1)\cdot k\\ \tilde{m}_2 &= 3\cdot(\sigma^2 + \tfrac{1}{2}\cdot(\mu_0^2+\mu_1^2))\\ \tilde{m}_3 &= \tfrac{3}{2}\cdot\sigma^2\cdot((\mu_0+\mu_1) + \tfrac{3}{2}\cdot(\mu_0^3+\mu_1^3))\cdot i\\ &\quad + \tfrac{3}{2}\cdot\sigma^2\cdot((\mu_0+\mu_1) + \tfrac{3}{2}\cdot(\mu_0^3+\mu_1^3))\cdot j\\ &\quad + \tfrac{3}{2}\cdot\sigma^2\cdot((\mu_0+\mu_1) + \tfrac{3}{2}\cdot(\mu_0^3+\mu_1^3))\cdot k.\end{aligned}$$

Then, $\hat{z}_0$ and $\hat{z}_1$ can be obtained by using (16) and (17):

$$\begin{aligned}z_{0v} &= \frac{-1}{6}\cdot(3\cdot c_{13} - \sqrt{9c_{13}^2 - 12c_{00}})\\ z_{1v} &= \frac{-1}{6}\cdot(3\cdot c_{13} + \sqrt{9c_{13}^2 - 12c_{00}})\end{aligned} \qquad \text{for } v = 1, 2, 3 \tag{19a}$$

where

$$\begin{aligned}c_{00} &= 3(\mu_0\cdot\mu_1 - \sigma^2)\\ c_{13} &= \frac{-(\mu_0+\mu_1)(\sigma^2 + \frac{3}{4}\cdot(\mu_0-\mu_1)^2)}{(3\cdot\sigma^2 + \frac{3}{4}\cdot(\mu_0-\mu_1)^2)}.\end{aligned} \tag{19b}$$

Suppose that two normal distributions are well separated with the assumption that $(\mu_0 - \mu_1) \gg \sigma$ and $(\mu_0\cdot\mu_1) \geq \sigma^2$. From (19b), c_{13} is approximated as

$$c_{13} \simeq -(\mu_0 + \mu_1).$$

Besides, we understand from (19a) that the values of z_{0v} and z_{1v} of (19a) are positive. This situation means that the input data set could be separated into two classes by the decision boundary l', which approximately passes through the point $(((\mu_0+\mu_1)/2), ((\mu_0+\mu_1)/2), ((\mu_0+\mu_1)/2)))$. On the other hand, when two normal distributions are not well separated, there might be only one class generated by the QMBTC. Considering the special case when two normal distributions are identical, that is $\mu_0 = \mu_1$, $\hat{z}_0$ and $\hat{z}_1$ become

$$\begin{aligned}z_{0v} &= \tfrac{1}{3}\cdot(\mu_0 - \sqrt{4\mu_0^2 + 9\sigma^2})\\ z_{1v} &= \tfrac{1}{3}\cdot(\mu_0 + \sqrt{4\mu_0^2 + 9\sigma^2})\end{aligned} \qquad \text{for } v = 1, 2, 3$$

from which we notice that z_{0v} is negative and the decision boundary l' passes through the point $((\mu_0/3), (\mu_0/3), (\mu_0/3))$. In this case, we can expect that QMBTC produces one output class with the quantized level $\hat{z}_1$ being approximately (μ_0, μ_0, μ_0) when σ is small enough.

In agreement with the examples in the last subsection, the above analysis reveals that QMBTC can decide the number of output classes according to the distribution of color values inside the pixel block. By this characteristic, QMBTC can achieve higher compression ratios since pixel blocks with similar color values need only one reproduction color, and the computed bit map can be replaced by a 1 bit reference indicating that one color clustering happened. However, the thresholding methods utilized by color BTC algorithms [7]–[9] always generate two output classes and no bit map saving.

IV. Application to Color Image Compression

In this section, we will introduce an approach of weighted color space to implement the QMBTC. To further compress the color image, a post-BTC data compression scheme, which combines the block merging and color codebook generation steps, is presented.

A. The Weighted Color Space Approach

As is known, the RGB space has extensive correlation among color components. Besides, pixels within the block are likely to have spatial correlation, which results in similar color values, except for the edge blocks. These two factors cause too many one-class pixel blocks by applying the QMBTC on the RGB space. Even though the compression ratio is high in this space, the picture quality of the reconstructed image is not acceptable from our empirical results. To alleviate this situation, we first transform the RGB space to the Yrg space.

$$\begin{aligned}Y &= \frac{(R+G+B)}{3}\\ r &= \frac{R}{(R+G+B)}\\ g &= \frac{G}{(R+G+B)}.\end{aligned}$$

In this space, the color components are decorrelated into two parts, the achromatic and chromatic components. The

$$X = \begin{pmatrix}(0,228,133,105) & (0,230,132,111) & (0,228,136,120) & (0,229,136,110)\\ (0,230,136,111) & (0,226,130,97) & (0,231,137,113) & (0,232,135,115)\\ (0,234,138,115) & (0,229,137,99) & (0,229,131,98) & (0,230,133,106)\\ (0,230,143,117) & (0,231,141,105) & (0,228,136,112) & (0,228,131,106)\end{pmatrix}$$

$$X' = \begin{pmatrix}(0,147,79,90) & (0,134,74,93) & (0,144,79,101) & (0,150,86,99)\\ (0,157,83,98) & (0,148,63,90) & (0,16,18,39) & (0,147,91,106)\\ (0,142,89,92) & (0,13,14,34) & (0,15,17,38) & (0,155,97,117)\\ (0,17,14,19) & (0,145,93,93) & (0,143,92,117) & (0,142,90,122)\end{pmatrix}$$

Fig. 1. Original testing color images referred to as (a) "Lena", (b) "Peppers", (c) "Scene", (d) "Jet".

achromatic Y component represents the intensity of the image and approximates the principal axis of the KL transform on the RGB space as reported by [15]. The r and g chromatic components represent the normalized chromatic information, respectively.

Then, the weights w_v for the associated color components $(I_1, I_2, I_3) = (Y, r, g)$ are determined by the smoothness of the Y values inside the pixel block. If the sample mean value of Y inside the pixel block is denoted as $\overline{Y}$, w_v are defined as

$$w_1 = 1 - \exp\left(-\frac{T}{\rho}\right)$$
$$w_2 = \frac{\exp\left(-\frac{T}{\rho}\right)}{2}$$
$$w_3 = \frac{\exp\left(-\frac{T}{\rho}\right)}{2}$$
$$T = \frac{1}{N}\sum_{n=1}^{N} |(Y(n) - \overline{Y}| \tag{20}$$

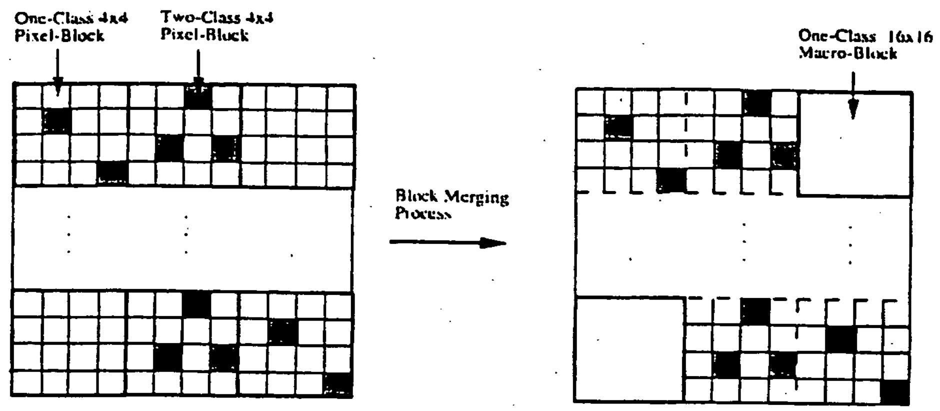
where ρ is a scaling factor and $Y(n)$ is the Y value of the nth pixel in the block. The weighted color components are thus $w_v \cdot I_v$, which are applied to the QMBTC. From (20), we see that T indicates the variation of Y values inside the pixel block. When $T = 0$, there is a constant Y value inside the pixel block. The CCC algorithm [7], which processes the Y component only, would not produce satisfactory results in this situation since there might be different colors existing within the pixel block. Nevertheless, the weighted color component approach of QMBTC can solve this problem. From (20), we understand that the weighted r and g components would dominate QMBTC, and would help it judge whether or not pixels inside the block should be truncated into two different classes.

TABLE I
THRESHOLDED BLOCK DISTRIBUTION OF QMBTC UNDER DIFFERENT COLOR SPACES

Images	Color Spaces	One-Class Blocks	Two-Class Blocks
LENA	I	11858	4526
	II	9408	6976
PEPPERS	I	12199	4185
	II	9408	6976
SCENCE	I	10396	5988
	II	7576	8808
JET	I	11430	4954
	II	9670	6714

Color Spaces : I -- RGB
II -- Weighted Yrg

In Table I, we illustrate the distribution of thresholded pixel blocks by using the QMBTC on test images shown in Fig. 1 when the input color spaces are the RGB space and weighted Yrg space with $\rho = 3.0$, respectively. It can be seen from Table I that the arrangement of w_v by (20) assists in improving the situation of too many one-class pixel



One-Class 4x4 Pixel-Block
Two-Class 4x4 Pixel-Block
Block Merging Process
One-Class 16x16 Macro-Block

Fig. 2. Block-merging stage of the post-BTC data compression scheme.

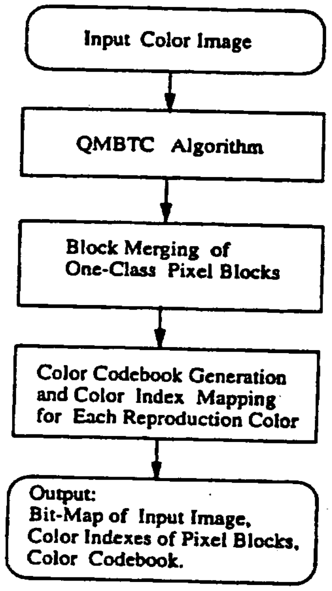
blocks for each test image. Besides, from empirical results, we noticed that the total computing time of QMBTC was enlarged about 6% by the weighted Yrg space approach on average.

B. Post-BTC Data Compression

To do color image compression, the proposed QMBTC is employed to each pixel block. After the number of classes has been decided, the centroid of each class is assigned as a reproduction color of the pixel block. The bit maps and the reproduction colors of all pixel blocks will then be sent to the receiving end and stored for decoding. However, since the total number of reproduction colors and bit maps for the whole image is still large, a post-BTC data compression scheme, which consists of two stages: 1) block-merging stage, and 2) color codebook generation stage, is presented in this paper to further compress the image.

In the block-merging stage as illustrated in Fig. 2, the input image will be first divided into nonoverlapping groups of 16 pixel blocks. In each group of pixel blocks, all blocks will be examined to see whether they can be merged as a larger pixel blocks, called a macroblock. If every block in the group is a one-class block, this group of blocks will be marked as a one-class macroblock. Otherwise, this group of blocks cannot be merged. For the case of 4×4 pixel block, the size of the merged macroblock becomes 16 × 16. For a merged macroblock, we then extract the average color value of each pixel block to form a 4×4 size block. On this block, QMBTC is executed to produce one or two reproduction colors for the entire merged macroblock. When one reproduction color is generated, the whole merged macroblock can be referenced with only one bit. Otherwise, when two reproduction colors are generated, each pixel block within a merged macroblock chooses its reproduction color according to its average color value belonging to the resultant color clustering.

Concerning the color codebook generation stage, a color quantization method proposed by Pei and Cheng [16], which partitions the color space of the image sequentially according to the order of color components, is first employed. Then, for each reproduction color generated by the QMBTC, its associated color index is generated by finding the nearest neighbor entry of the color codebook. The proposed post-BTC



Input Color Image
QMBTC Algorithm
Block Merging of One-Class Pixel Blocks
Color Codebook Generation and Color Index Mapping for Each Reproduction Color
Output: Bit-Map of Input Image, Color Indexes of Pixel Blocks, Color Codebook.

Fig. 3. Process of the post-BTC data compression scheme.

data compression process is shown in Fig. 3. To decode each compressed pixel block, the bit map, color indexes of each pixel block, and the color codebook are the only required elements. The receiver can display the decoded data through a lookup table manner in real time.

V. Experimental Results

In this section, we conducted the experiments on images of Fig. 1, called "Lena," "Peppers," "Scene," and "Jet." These test color images with 512 × 512 size are coded at 8 bits/component. The average peak signal-to-noise-ratio (APSNR) criterion described as follows is used to judge the picture quality.

$$\mathrm{APSNR} = 10 \log \frac{3 * N * 255^2}{\mathrm{TSE}} \tag{21}$$

where

$$\mathrm{TSE} = \sum_{\text{all pixels}} \|I - Q(I)\|^2$$

TABLE II
PERFORMANCE COMPARISONS OF VARIOUS COLOR BTC ALGORITHMS

Images	Algorithms	PSNR(R) (dB)	PSNR(G) (dB)	PSNR(B) (dB)	APSNR (dB)	Bit Rate (bits / pixel)
LENA	I	33.0	31.6	32.1	32.2	4.0
	II	33.1	31.6	32.5	32.3	4.0
	III	32.9	31.5	31.9	32.0	2.56
	IV	34.5	36.1	31.8	33.8	2.46
PEPPERS	I	31.9	30.1	31.5	31.1	4.0
	II	31.8	30.8	32.3	31.6	4.0
	III	31.2	30.1	31.2	30.8	2.56
	IV	30.1	33.4	30.6	31.2	2.54
SCENCE	I	30.2	27.1	27.6	28.1	4.0
	II	30.1	27.4	27.8	28.2	4.0
	III	30.0	27.2	27.9	28.2	2.84
	IV	28.7	31.1	28.1	29.1	2.80
JET	I	31.7	30.6	33.8	31.8	4.0
	II	31.8	30.8	33.9	32.0	4.0
	III	31.9	31.1	34.2	32.2	2.52
	IV	35.1	34.4	33.3	34.2	2.51

Methods : I – CCC Algorithm
II – Kurita and Otsu's Algorithm
III – the Proposed QMBTC Algorithm
IV – the JPEG standard

TABLE III
PERFORMANCE OF QMBTC AFTER POST-BTC DATA COMPRESSION

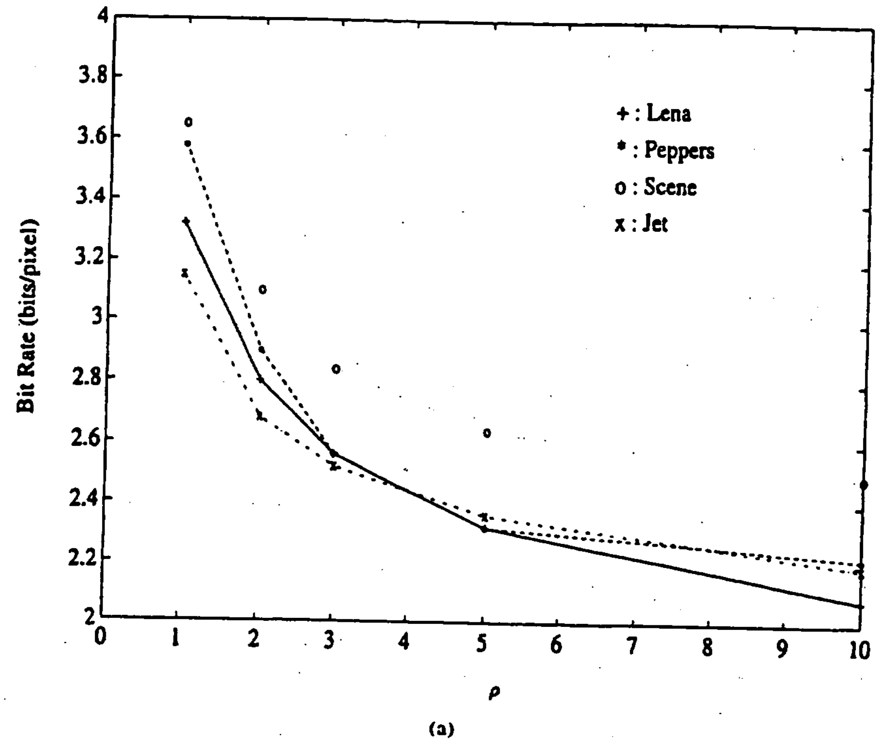
Images	PSNR(R) PSNR(G) PSNR(B) (dB)	APSNR (dB)	Bit Rate (bits / pixel)
LENA	31.7 30.7 31.0	31.1	1.11
PEPPERS	29.5 29.0 29.8	29.4	1.10
SCENCE	28.9 26.7 27.3	27.5	1.28
JET	31.3 32.6 33.3	31.6	1.02

with I and $Q(I)$ representing the original color values and the quantized color values after color BTC, respectively. In the following empirical results, we express the color values I and $Q(I)$ on the RGB space. Fig. 4(a) and (b) shows how the scaling factor ρ of (20) changes the bit rate (bits/pixel) and APSNR, respectively, when the coded color images were not passed through the post-BTC data compression scheme. Fig. 4(a) plotted the bit rate performance with ρ equal to 1, 2, 3, 5, and 10 for four test images. On the other hand, Fig. 4(b) shows the APSNR performance for various ρ values. From these results, we observe that the bit rate and APSNR monotonically decrease with the ρ value. We also notice that $\rho = 3$ is a good compromise between bit rate and APSNR performance. Therefore, $\rho = 3$ is selected in the following experiments of the proposed QMBTC.

For the purpose of performance evaluation, we compare the QMBTC without post-BTC data compression with two existing color BTC algorithms, which are: 1) the CCC algorithm, and 2) Kurita and Otsu's algorithm. With the same 4×4 pixel block size, Table II illustrates the performance comparison among three testing color BTC algorithms. The separate PSNR's for each component, PSNR(R), PSNR(G), PSNR(B), are also included. In particular, we observe that the bit rate will save 30% on average by the proposed QMBTC algorithm as compared with the other testing algorithms. Nevertheless, the APSNR of the proposed algorithm is close to those of the other testing algorithms. In this table, we also include the performance of the JPEG international standard coding scheme [17] when its bit rate is close to QMBTC. The JPEG standard is based on the discrete cosine transform (DCT) to compress each component of the color image. Using the JPEG standard to compress an input color image, we transformed the input color coordinate from the RGB space to the CCIR601-YCrCb space, which was then downsampled two times for each chromatic component. To calculate the APSNR of the reconstructed image, we did the inverse operation of the above downsampling and color transformation in order to have equal picture size for each component of the RGB space. As we see, the APSNR of the JPEG standard is about 1 dB higher than that of QMBTC on average. However, we observed from experimental results that reconstructed images of the JPEG standard and QMBTC are essentially indistinguishable. The advantage of QMBTC is in the low decoder complexity. It is hard to decode the JPEG compressed image in real time without the assistance of specific hardware to do the inverse-DCT operation.

Figs. 5 and 6 show the reconstructed images created by using these color BTC algorithms on "Lena" and "Jet," respectively. The original image of "Lena" is shown in Fig. 5(a). The reconstructed image of "Lena" from reproduction colors of pixel blocks by the proposed algorithm is shown in Fig. 5(b). The distribution of thresholded pixel blocks produced by using the proposed QMBTC is illustrated in Fig. 5(c), in which black represented a one-class pixel block and white represented a two-class pixel block. Fig. 5(d) and (e) represents the results for the other testing algorithms. Fig. 6 shows a similar simulation result for "Jet." It is noticed that there is no significant image quality degradation between the reconstructed images produced by the proposed algorithm and those of the other testing algorithms.

Finally, Table III compares the APSNR's and bit rates of the proposed QMBTC algorithm for four test images after being processed by the post-BTC data compression scheme. The codebook generated by [16] consists of 256 colors. In order to reduce the blocking effect of a one-class macroblock, we have checked the sample mean value of each generated one-class macroblock, and have observed whether it was located within the range $[T_L, T_H]$. The values of T_L and T_H were empirically chosen as 120 and 200, respectively. If this threshold range was met, the status being the one-class macroblock was canceled. As Table



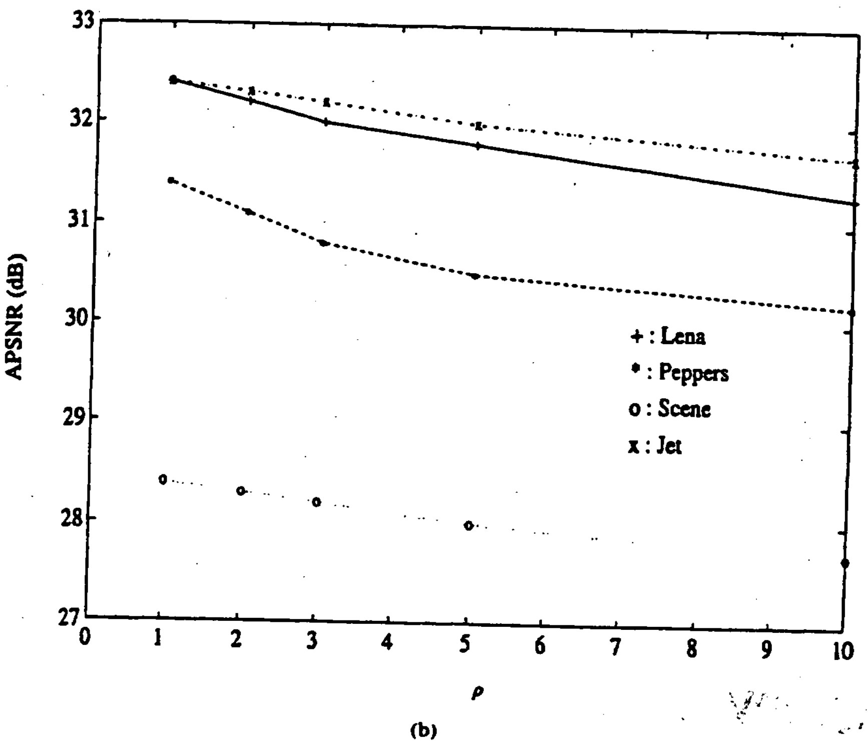


Fig. 4. (a) Output bit-rate distribution of the QMBTC versus the scaling factor ρ for four test images. (b) Output APSNR distribution of the QMBTC versus the scaling factor ρ for four test images.

III shows, the bit rate can be further decreased to 1.13 bits/pixel on average. In addition, from empirical results, we noticed that the total computing time of QMBTC was enlarged about 3% by post-BTC data compression on average. The associated reconstructed images for "Lena", "Peppers", "Scene", and "Jet" are shown in Fig. 7. Although some blocking artifacts exist on the reconstructed images f "Lena" and "Peppers," their picture quality is acceptable to the human eye at a distance of four–five times the picture height.

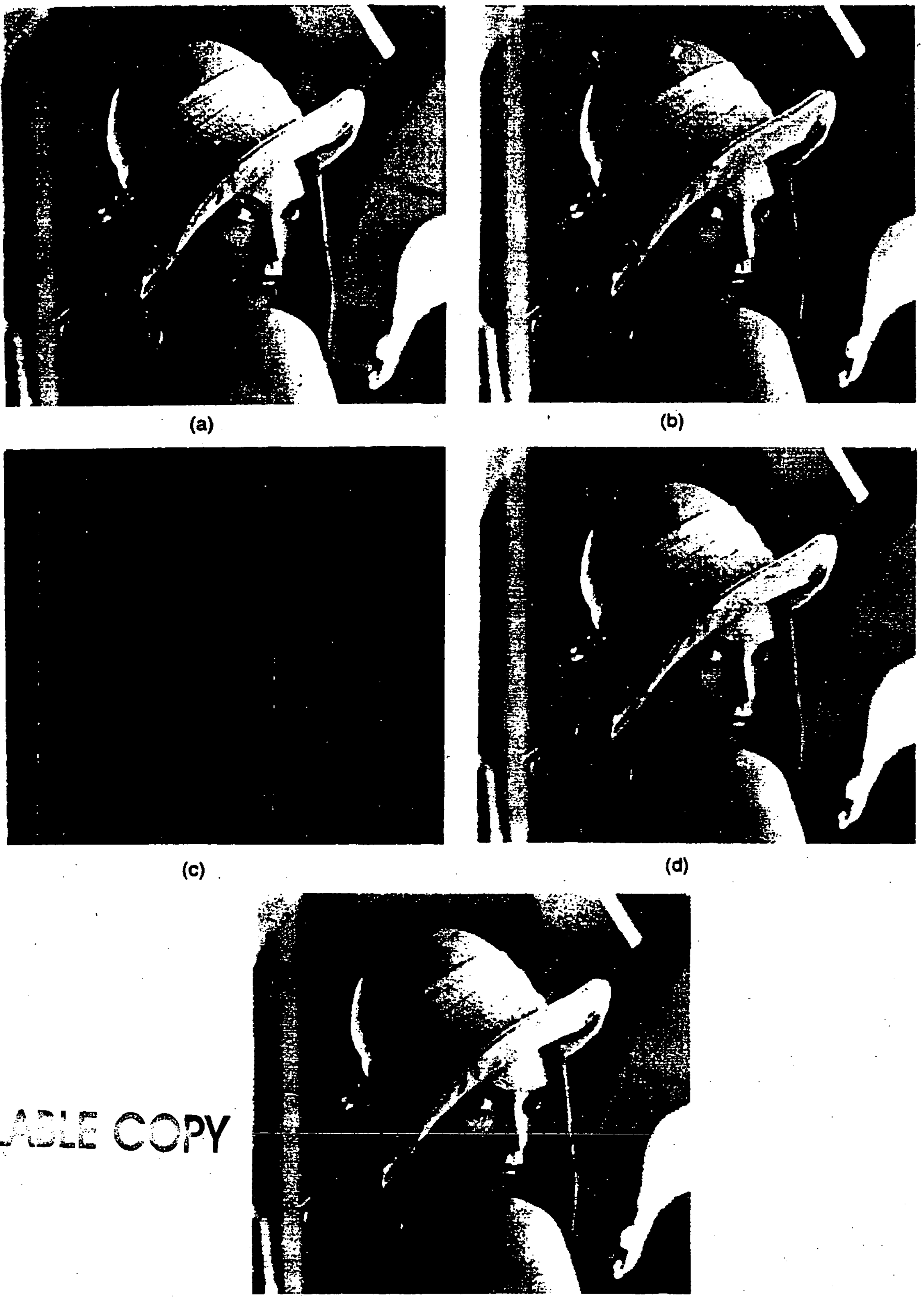


Fig. 5. Original and reconstructed images of "Lena." (a) Original image. (b) Reconstructed image by the QMBTC with block size 4 × 4. (c) Thresholded block distribution by the QMBTC. (d) Reconstructed image by the CCC algorithm with block size 4 × 4. (e) Reconstructed image by the Kurita and Otsu algorithm with block size 4 × 4.

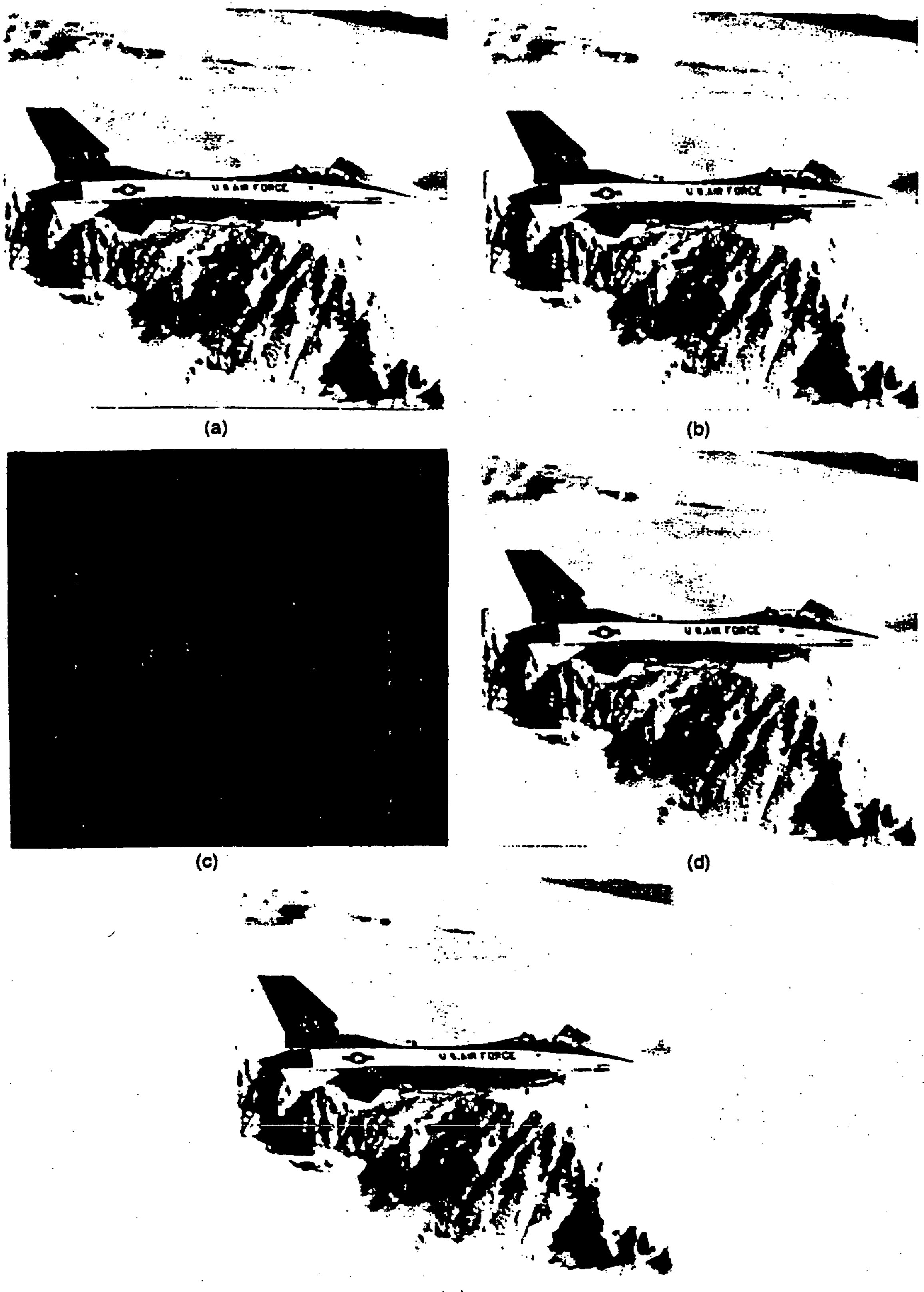


Fig. 6. Original and reconstructed images of "Jet." (a) Original image. (b) Reconstructed image by the QMBTC with block size 4 × 4. (c) Thresholded block distribution by the QMBTC. (d) Reconstructed image by the CCC algorithm with block size 4 × 4. (e) Reconstructed image by the Kurita and Otsu algorithm with block size 4 × 4.

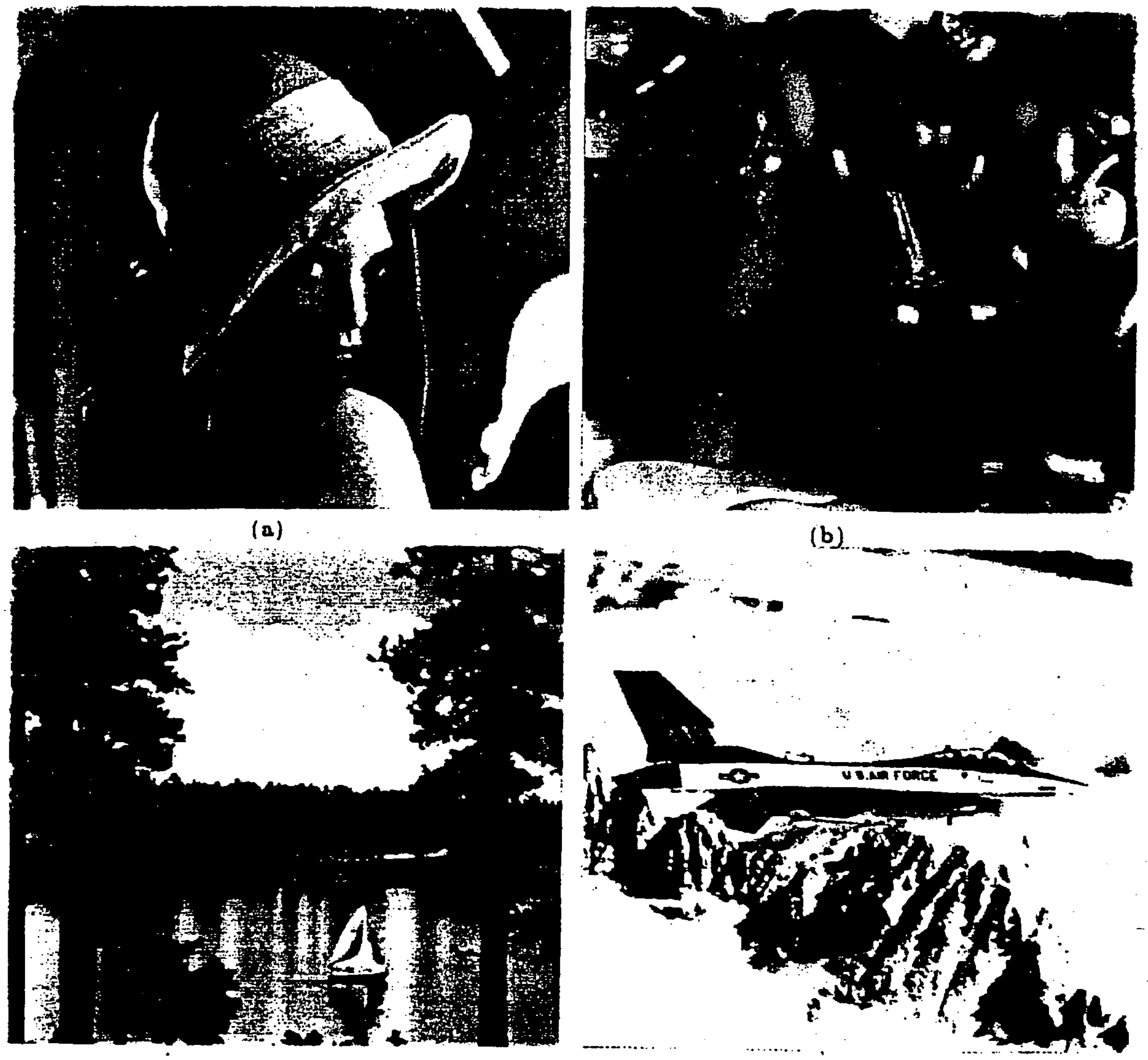


Fig. 7. Reconstructed images of QMBTC after post-BTC data compression. (a) Reconstructed image of "Lena." (b) Reconstructed image of "Peppers." (c) Reconstructed image of "Scene." (d) reconstructed image of "Jet."

VI. Conclusions

In this paper, we present a color BTC algorithm, called the QMBTC, to perform color image compression. The proposed algorithm can produce one-class or two-class pixel blocks through quaternion-moment-preserving thresholding. From the derived formulas, we know that the computation complexity of the QMBTC is of order N, the size of pixel block. Empirical results, which compare the APSNR's and bit rates of the proposed algorithm with those of existing single bit map color BTC algorithms, show that a 30% saving is achieved on average by the proposed algorithm, while no significant degradation of picture quality is perceived. To further compress color image, a post-BTC data compression scheme, which combines the block-merging and color codebook generation steps, is employed. The bit rate is then dropped to 1.13 bits/pixel on average with acceptable output picture quality.

Acknowledgment

The authors would like to thank the anonymous reviewers who made many useful comments. Their help is gratefully appreciated.

References

[1] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Commun.*, vol. COM-27, pp. 1335–1341, Sept. 1979.
[2] O. R. Mitchell and E. J. Delp, "Multilevel graphics representation using block truncation coding," *Proc. IEEE*, vol. 68, July 1980.
[3] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood, NJ: Prentice-Hall, 1991.
[4] M. D. Lema and O. R. Mitchell, "Absolute moment block truncation coding and application to color image," *IEEE Trans. Commun.*, vol. COM-32, pp. 1148–1157, Oct. 1984.
[5] V. R. Udpikar and J. P. Raina, "BTC image coding using vector quantization," *IEEE Trans. Commun.*, vol. COM-32, pp. 1148–1157, Oct. 1984.

[6] D. J. Healy and O. R. Mitchell, "Digital video bandwidth compression using block truncation coding," *IEEE Trans. Commun.*, vol. COM-29, pp. 1809–1817, Dec. 1981.
[7] G. Campbell, T. A. Defanti, J. Frederiksen, S. A. Joyce, A. L. Lawrence, J. A. Lindberg, and D. J. Sandin, "Two bit/pixel full color encoding," *Comput. Graph.*, vol. 20, pp. 215–223, Aug. 1986.
[8] Y. Wu and D. C. Coll, "Single bit-map block truncation coding of color images," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 952–959, June 1992.
[9] T. Kurita and N. Otsu, "A method of block truncation coding for color image compression," *IEEE Trans. Commun.*, vol. COM-41, pp. 1270–1274, Sept. 1993.
[10] R. Machuca and K. Phillips, "Applications of vector fields to image processing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 3, pp. 318–329, 1983.
[11] J. B. Fraleigh, *A First Course in Abstract Algebra*. Reading, MA: Addison-Wesley, 1982.
[12] C. K. Papadopoulos and C. L. Nikias, "Parameter estimation of exponentially damped sinusoids using high order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-38, pp. 1424–1435, Aug. 1990.
[13] S. Z. Selim and M. A. Islam, "K-means type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 81–87, 1984.
[14] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 1984.
[15] Y. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Comput. Vision, Graph., Image Processing*, vol. 13, pp. 222–241, 1980.
[16] S. C. Pei and C. M. Cheng, "Dependent scalar quantization of color images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. CSVT-5, pp. 124–139, Apr. 1995.
[17] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, pp. 30–44, Apr. 1991.

Soo-Chang Pei (S'71–M'86–SM'89) was born in Soo-Auo, Taiwan in 1949. He received BSEE from National Taiwan University in 1970 and MSEE and Ph.D. from University of California, Santa Barbara in 1972 and 1975 respectively.

He was an engineering officer in the Chinese Navy Shipyard from 1970 to 1971. From 1971 to 1975, he was a research assistant at the University of California, Santa Barbara. He was Professor and Chairman in the EE department of Tatung Institute of Technology from 1981 to 1983. Presently, he is the Professor and Chairman of EE department at National Taiwan University. His research interests include digital signal processing, image processing, optical information processing, and laser holography. Dr. Pei is member of the IEEE, Eta Keppa Nu and the Optical Society of America.

Ching-Min Cheng (S'92–M'95) was born in Taipei, Taiwan, in 1959. He received the BSEE from National College of Marine Science and Technology, Keelung, Taiwan, in 1982 and MSEE from University of California, San Diego, in 1986. In 1996, he received the Ph.D. degree in electrical engineering from National Taiwan University.

From 1983 to 1984 he was an engineering officer in the Chinese Airforce Anti-aircraft Corps. From 1986 to Aug. 1989 he served as a patent examiner in the National Bureau of Standards. Since Sep. 1989, he has joined Telecommunication Labs, Ministry of Communications, Taiwan as a research engineer. His research interests includes digital signal processing, video compression, and multimedia communication.